Adaptation of EAP-NOOB Method for LoRaWAN with LO-CoAP-EAP and CBOR

Eduardo Ingles Sanchez, Dan Garcia-Carrillo, Georgios Papadopoulos, Nicolas Montavont, Antonio F Skarmeta Gómez

Adaptation of EAP-NOOB Method for LoRaWAN with LO-CoAP-EAP and CBOR

Eduardo Ingles-Sanchez[†,*], Dan Garcia-Carrillo[‡], Georgios Z. Papadopoulos[*], Nicolas Montavont[*] and Antonio F. Skarmeta Gómez[†]

[†] University of Murcia, Spain, Email: eduardo.ingles@um.es, skarmeta@um.es
[*] IMT Atlantique, Irisa, France, Email: {firstname.lastname}@imt-atlantique.fr
[‡] Odin Solutions SL, Murcia, Spain, Email: dgarcia@odins.es

***Abstract*—The Internet of Things (IoT) is driving research and innovation in the areas of security and device management. The bootstrapping procedure is yet another security problem that needs to be solved. At the bootstrapping period, a newly deployed device is performing a set of actions that allows it to join a network as a trusted party. One of the currently proposed methods in the Internet Engineering Task Force (IETF) EAP Method Update (EMU) Working Group (WG) is employing the Extensible Authentication Protocol (EAP) to enable the authentication of IoT devices in more efficient and scalable ways. In this sense, we select and investigate the suitability one of the newly proposed EAP method in LoRaWAN, a Low-Power Wide-Area Network (LPWAN) technology. EAP-Nimble out-of-band (EAP-NOOB) method works without pre-established configuration and enables an out-of-band channel to improve authentication. In this paper, we analyse how the EAP-NOOB method can be employed in LoRaWAN networks, we then propose the use of two mechanisms that can be used to reduce the message size of the EAP-NOOB exchange.**

***Index Terms*—EAP, Authentication, IoT, EAP-NOOB, Bootstrapping, Enrollment, IoT onboarding, LoRaWAN, LPWAN, CoAP-EAP, LO-CoAP-EAP**

I. Introduction

Security in the Internet of Things (IoT) is an ongoing subject of research and development by the research institutions, manufacturers and standardisation organisations. Among the different approaches that tackle IoT security, bootstrapping is the process that allows a secure joining for new IoT devices to a deployed and operating network. In this sense, a typical approach in bootstrapping is to provide a tailored solution to the technology, which opens the door to the development of solutions that enable interoperabilty, regardless of the technology being used in IoT.

Some of the efforts in this line are done in the context of the Extensible Authentication Protocol (EAP) [1]. This is due to the possibility of running—virtually—any authentication method and having the support of Authentication Authorisation and Accounting (AAA) infrastructures [2], which are currently being used by Telcos go provide support to mobile phones subscribers. The efforts can go towards providing a lightweight alternative to the existing related protocols (e.g., developing lightweight EAP lower layers), or providing a more scalable solution to ease the process of deployment of the devices. The EAP-Nimble out-of-band (EAP-NOOB) method aims to the later, providing a suitable alternative to IoT deployments during the bootstrapping period. Relieving thus the deployment of having to program each device with unique credentials.

The reminder of this article is as follows: Section II provides the necessary technical background. Section III provides the rationale for the work, presenting the starting point of the work. Section IV presents the proposal, while in Section V, we elaborate the conclusions and future work.

II. Background & Technical Overview

A. CoAP-EAP and LO-CoAP-EAP

Once an IoT device is deployed and powered on, it should perform a bootstrapping process to become a trusted party on the network and, thus, part of the security domain. The bootstrapping process entails running security procedures such as authentication, authorisation and key management. This process is of considerable importance for operators that have to manage large numbers of IoT devices in IoT deployments. Bootstrapping should ensure that it is lightweight, based on standard protocols and is interoperable to facilitate a standalone solution to any IoT technology.

Constrained Application Protocol EAP (CoAP-EAP) is a promising solution to tackle with the previously presented issue, an EAP lower layer implementation that is designed considering the constrains of IoT [3]. CoAP-EAP provides a lightweight CoAP-based bootstrapping service for the IoT. CoAP-EAP leverages three current standards to provide a lightweight service:

1) CoAP, a lightweight Representational State Transfer (REST)-based web transfer protocol, employed for the bootstrapping, and for the EAP lower layer design.
2) EAP, an extensible protocol with native integration of AAA that supports a large number of authentication methods. It includes new EAP methods currently considered in the Internet Engineering Task Force (IETF) EAP Method Update (EMU) Working Group (WG).
3) AAA is used in massive deployments, where user identity management is typically required, such as the ones from Telcos for mobile phone subscribers authentication.

Another use case is the Eduroam, where users from universities and research institutions, within Eduroam, can have Internet access.

LPWAN WG discussed the deployment of LPWAN networks in conjunction with AAA infrastructures for access authentication. Since LPWAN is similar to existing cellular deployments but with constrained resources, it can be a candidate for massive deployments of devices such as IoT [4].

A step further in the optimisation of the EAP lower layer for IoT, is the redesign of CoAP-EAP[3]. Low-Overhead CoAP-EAP (LO-CoAP-EAP) [4], is a redesign of CoAP-EAP that considers the limitations of LPWAN. It reduces not only the size of the messages, but also the messages that are optional according to the EAP standard.

Having in mind that we investigate the possibility of using a new EAP method in LPWAN, LO-CoAP-EAP will be considered for this purpose. LO-CoAP-EAP allows to run any EAP method by reducing the overhead of current EAP lower layers. This solution is more suitable for very constrained IoT environments, unlike other proposals such as Protocol for Carrying Authentication for Network Access (PANA), which is the current standard proposed as EAP lower layer for IoT.

B. EAP-NOOB Method

The extensive use of EAP for bootstrapping turns it into an interesting case of study by providing a solution for all kind of devices; including IoT devices. Among all the EAP methods, EAP-NOOB [5] provides some exciting properties that make it a suitable solution for IoT.

The network enrollment with EAP-NOOB allows a device or EAP peer to establish a security association without authentication. Constrained devices do not need any pre-configured information. In terms of security, we do not pre-establish any identifier or security credentials. It means that the devices do not depend on any manufacturer or third party.

This method exchanges a shared secret key between the peer and server by using an Elliptic-curve Diffie–Hellman (ECDH) key agreement protocol.

A particular feature of this authentication algorithm is that it includes an Out-Of-Band (OOB) channel to ensure that security information is not sent through a single channel. A channel assisted with a third actor sends a nonce called Noob and a cryptographic fingerprint (Hoob) from the peer to the server or backwards. To carry out this procedure, the constrained device must contain at least one input or output interface (e.g. camera, screen, blinking LED).

EAP-NOOB is divided into several phases, where each phase is an EAP conversation. Thus, there are multiple EAP conversations:

- *Initial Exchange*: It initiates a handshaking process to negotiate the configuration parameters. The device and the server exchange the peer identifier for the current and future EAP conversations. They exchange nonces, and the ECDH exchange is performed to provide a shared secret. This EAP conversation intentionally ends with an EAP-Failure.
- *Waiting Exchange*: Once the association between the device and the server is created, a third party, typically the user, must complete the OOB step. The protocol awaits a particular time and tries to reconnect with the server using EAP-NOOB. If the out-of-band procedure is completed, it will start with the *Completion Exchange*. Otherwise, the communication ends with an EAP-Failure; the device waits again until the timeout expires and repeats this exchange.
- *Completion Exchange*: Here, both entities exchange message authentication codes to verify that the shared cryptographic materials are trusted. As a result, if the information is reliable, the conversation ends with an EAP-Success, and both devices maintain a persistent EAP-NOOB association.

It also implements a *Fast Reconnect Exchange* to avoid repeating OOB step again [5]. It allows a device to reconnect by using the cryptographic material from a previous association.

C. Concise Binary Object Representation (CBOR)

The Concise Binary Object Representation (CBOR), specified in [6], is a data format that allows obtaining a small message, while having a considerable small code size.

CBOR uses the JSON data model as its underlying data model and has a set of goals such as:

1) Unambiguously encode the formats most commonly used in Internet standards.
2) It should be usable in constrained devices with very limited memory and processing power.
3) The date should be decoded without the need of a schema description.
4) Obtain a compact serialisation.
5) It should be usable in constrained and non-constrained applications.
6) Should be compatible with JSON, to convert CBOR to and from JSON.
7) Must be extensible.

Hereafter, we provide a simple example that demonstrates the capabilities of CBOR.

Listing 1. Example of JSON document

```
{
  "name":"Variable Name",
  "data":"0123456789"
}
```

The previous JSON Object example has a 44 Bytes size.

Listing 2. Example of CBOR from the JSON document in Listing 1

```
A2                      # Map(2)
   64                   # Text(4)
      6E616D65          # "name"
   6D                   # Text(13)
      5661726961626
      C65204E616D65     # "Variable Name"
   64                   # Text(4)
      64617461          # "data"
   6A                   # Text(10)
      3031323334
      3536373839        # "0123456789"
```

The CBOR format has a size of 36 Bytes, which implies to 18.18% of a reduction, in this simple instance. This is a potential tool to reduce bytes sent over the air when JSON is the current format in use.

D. LoRaWAN

The increasing proliferation of devices with limited memory and battery consumption led to the emergence of several Low-Power Wide-Area Networking (LPWAN) technologies [7]. In this sense, the creation of IETF LPWAN WG has the aim of standardising protocols that allow the use of CoAP, UDP and IPv6 packet over LPWAN networks.

LPWAN technologies allow transmission of packets of up to several kilometres with battery-powered devices. The scientific community and industry, supported by companies like Semtech, have good acceptance of LoRaWAN, an LP-WAN technology. It uses unlicensed frequency bands. Its link-layer protocol is based on LoRa module, while LoRaWAN defines the Medium Access Control (MAC) over its physical layer [8], [9]. Due to the low-transmission rates, LoRaWAN imposes further limitations. It allows sending messages with a payload of up to 51 Bytes or 222 Bytes in the best conditions, depending on the Spreading Factor (SF) [10], [11], [12].

LPWAN WG members work on optimising CoAP with protocols like SCHC [13]. However, there are also efforts dedicated to the integration of AAA Infrastructures protocols (e.g. RADIUS [14]). The use of CoAP-EAP in conjunction with EAP-NOOB brings together the efforts to provide a solution that meets the requirements of an AAA Infrastructure.

III. Problem Statement

Following the efforts of providing adequate levels of security to the massive deployments in IoT, we evaluate the suitability of EAP-NOOB method, one of the ongoing efforts of the IETF EMU Working Group. We evaluate EAP-NOOB for use in LPWAN IoT technologies and, more specifically, we employ the LoRaWAN technology. The current work on EAP-NOOB is evaluated using the theoretical values from the draft [5] and the values from the works of CoAP-EAP [3] and LO-CoAP-EAP[4]. We first evaluate the work on CoAP-EAP as a baseline, since the structure of the exchanges resembles a canonical EAP exchange which considers EAP REQ/ID and EAP REP/ID. After evaluating its use with CoAP-EAP, we will contrast these values with the use of LO-CoAP-EAP instead. We further explore optimisations in the message format of the EAP-NOOB method with CBOR, replacing thus the JSON content with a more concise representation of the exchanged information.

A. EAP-NOOB over CoAP-EAP

In this section, we show the operation flow of EAP-NOOB carried over CoAP-EAP. In Fig. 1, the exchanged messages are depicted, while in Table I, we list the message size.

The communication entails 18 exchanged messages, divided in 2 phases. The first phase, called *Initial Exchange*, runs on steps **1-10**, and the second phase, called *Completion Exchange*, runs on steps **11-18**. In each phase, there are two differentiated stages. First, a CoAP association between the Smart Object with the Controller and, secondly, an EAP conversation with the server.

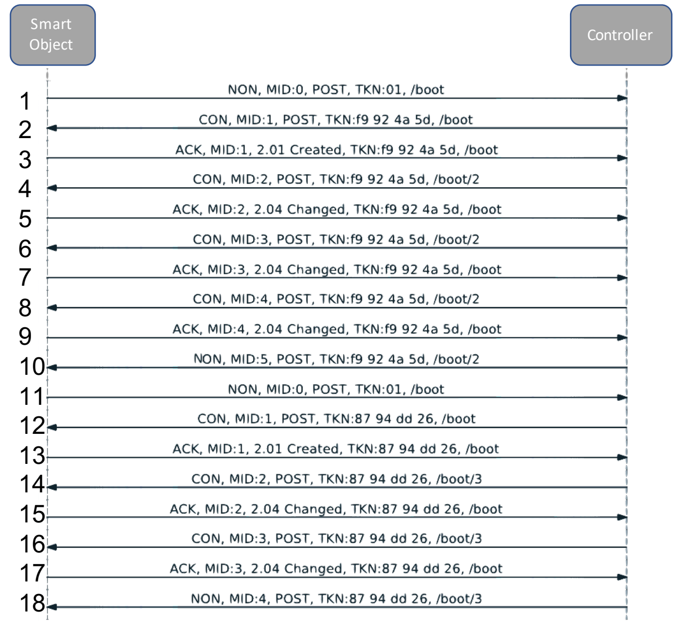

Fig. 1. CoAP-EAP flow for EAP-NOOB

The first three messages (steps **1-3**) enables the association of the client with the CoAP Controller. This process is repeated in steps **11-13** due to EAP-NOOB multiple EAP conversations. After the initial trigger, CoAP-EAP follows the systematic process of an EAP exchange, sending the EAP Request/Identity to the Smart Object, which responds to the Controller with the EAP Response/Identity (steps **4-5**). This message is then sent to the EAP server, which chooses the EAP method to be used. For the sake of simplicity, the exchange with the EAP server is not shown in this case. Thus, we observe that the beginning of each conversation requires 5 message exchanges (steps **1-5** and **11-15**).

During the Initial Exchange, the peer and the server agree on configuration parameters (steps **6-7**). It includes the information needed for the ECDH exchange in steps **8-9**. Later, the Smart Object initiates a new connection with the server. In this example, we assume that the OOB step has been completed on time between the step **10** and step **11**. Therefore, it sends the corresponding message for the *Completion Exchange* phase (steps **16-17**). Eventually, the data is validated locally and the exchange is successfully completed in step **18**. In case the OOB step is not carried out on time, steps **16-17** would be part of *Waiting Exchange* indicating that it should await more. Step **18** would end up in an EAP-Failure.

B. CoAP-EAP Message Sizes

The last column in Table I shows the size in Bytes of each CoAP-EAP message. The messages that initiate the EAP conversation (1-5 and 11-15), as well as the messages that return the EAP state (10, 18) take less than 51 Bytes each. Therefore, there is no need for optimisation for a LoRaWAN network. Nevertheless, CoAP-EAP requires 101 Bytes (1-5) and 126 Bytes (11-15), respectively, to start the conversation. The remaining messages in the Table include specific data for the EAP-NOOB method. Each EAP-NOOB message includes the EAP header and a text in JSON format. Considering the limitations of LoRaWAN, message size of 8 and 9 are beyond the limitations of the specification. Besides, the large size of messages 6, 7, 16 and 17 can also degrade the network performance. Note that EAP-NOOB relies on timers during the authentication process. Therefore, reducing the message Time-on-Air (ToA) may become the crucial factor that allows authentication to be done without taking too much time.

TABLE I
EAP-NOOB MESSAGE SIZE WITH COAP-EAP AND LO-COAP-EAP

Nº	CoAP message type	EAP-NOOB message type	With LO-CoAP-EAP (Bytes)	With CoAP-EAP (Bytes)
1	NON POST	Client init	27	10
2	CON POST	Controller init	-	18
3	ACK POST	Ack CoAP init	-	20
4	CON POST	EAP ID req	-	21
5	ACK POST	EAP ID resp	-	32
6	CON POST	Type 1	196	203
7	ACK POST	Type 1	150	154
8	CON POST	Type 2	258	265
9	ACK POST	Type 2	239	243
10	NON POST	EAP Failure	13	20
11	NON POST	Client init	52	10
12	CON POST	Controller init	-	18
13	ACK POST	Ack CoAP init	-	20
14	CON POST	EAP ID req	-	21
15	ACK POST	EAP ID resp	-	57
16	CON POST	Type 4	137	153
17	ACK POST	Type 4	103	112
18	NON POST	EAP Success	4	20
		TOTAL	**1202**	**1397**

IV. ADAPTATION OF EAP-NOOB METHOD FOR LORAWAN

As we presented in the previous section, some of the EAP-NOOB messages are too long to be employed in LoRaWAN. Thus, there is room for optimising the protocol and, thus, reducing the number of bytes sent Over-The-Air. In this paper, we propose two optimisation strategies:

- Using LO-CoAP-EAP as EAP lower layer.
- Using CBOR to reduce the JSON Objects sent in the EAP-NOOB messages.

A. Transporting EAP-NOOB over LO-CoAP-EAP

The use of LO-CoAP-EAP allows us to reduce the size of the EAP lower layer used. Some of the improvements, as explained in the original work are:

1) Reduce the URI length that represents the bootstrapping service to the lowest instance.
2) Eliminate EAP messages that are not mandatory: EAP Request and Response Identity.
3) Reduce Token to the minimum instance: 0 bytes.
4) Embed nonce values in other messages instead of using specific messages.

Table I shows the result of applying the reductions proposed above. As an example, we see that to initiate the conversation (the trigger message *Client init*), LO-CoAP-EAP requires only one message to send as opposed to the 5 messages that CoAP-EAP transmits. This is because LO-CoAP-EAP gathers the nonce and EAP-Identity in the same message.

B. Reducing the EAP-NOOB content with CBOR

Converting JSON objects to CBOR format offers the possibility to reduce the size of the EAP messages exchanged in EAP-NOOB. Listing 3 illustrates an example of EAP-NOOB Type 1 message in JSON object format. The size of the object is 181 Bytes.

Listing 3. Example of Type 1 EAP-NOOB in JSON.

```
{
 "Type": 1,
 "Vers": [1],
 "PeerId": "vnybKEHlpmuhSP8dJaclXD",
 "Cryptosuites": [1],
 "Dirs": 3,
 "ServerInfo": {
  "Name": "Example",
  "Url": "https://localhost:8080/sendOOB"
 },
 "Realm": "noob.example.com"
}
```

The new size of the JSON object after converting to CBOR format is 149 Bytes. In the process, we gain a reduction of 17.68%. Listing 4 shows the result parsed for the convenience of the reader.

Listing 4. Example of Type 1 EAP-NOOB message content after applying CBOR.

```
A7                      # map(7)
    64                  # text(4)
        54797065        # "Type"
    01                  # unsigned(1)
    64                  # text(4)
        56657273        # "Vers"
    81                  # array(1)
        01              # unsigned(1)
    66                  # text(6)
        506565724964    # "PeerId"
    76                  # text(22)
        766E79624B45
        486C706D7568
        535038644A61
        636C5844    # "vnybKEHlpmuh
```

```
                   #   SP8dJaclXD"
6C                    #  text(12)
   43727970746F
   737569746573    #  "Cryptosuites"
81                    #  array(1)
   01                 #  unsigned(1)
64                    #  text(4)
   44697273           #  "Dirs"
03                    #  unsigned(3)
6A                    #  text(10)
   536572766572
   496E666F   #  "ServerInfo"
A2                    #  map(2)
   64                 #  text(4)
      4E616D65        #  "Name"
   67                 #  text(7)
      4578616D
      706C65          #  "Example"
   63                 #  text(3)
      55726C          #  "Url"
   78  1E             #  text(30)
      68747470
      733A2F2F
      6C6F6361
      6C686F73
      743A3830
      38302F73
      656E644F
      4F42  #  "https://localhost:8080/
            #   sendOOB"
65                 #  text(5)
   5265616C6D#  "Realm"
70                 #  text(16)
   6E6F6F622E
   6578616D70
   6C652E636F6D    #  "noob.example.com"
```

TABLE II
EAP-NOOB MESSAGE SIZE AFTER APPLYING LO-COAP-EAP AND CBOR TO THE JSON CONTENT OF EAP-NOOB

N	Message	Content (LO-CoAP-EAP + EAP-NOOB)	Length (Bytes)
1	NON POST	Client trigger with nonce and Identity	27
2	CON POST	Type 1 - JSON - CBOR	158
3	ACK	Type 1 - JSON - CBOR	113
4	CON POST	Type 2 - JSON - CBOR	222
5	ACK	Type 2 - JSON - CBOR	206
6	NON POST	EAP Failure	13
7	NON POST	Client trigger with nonce and Identity	52
8	CON POST	Type 4 - JSON - CBOR	126
9	ACK	Type 4 - JSON - CBOR	92
10	NON POST	EAP Success	13
	TOTAL		1022

TABLE III
LORAWAN PARAMETERS TO GET THE TIME-ON-AIR OF THE EXCHANGE

Parameter	Value
LoRaWAN Header size	13
Explicit header	yes
Low DR optimization	auto
Coding rate	4/5
Preamble symbols	8
Bandwidth	125kHz
Spreading Factor	SF7

TABLE IV
TIME-ON-AIR OF THE EXCHANGE

Parameters	Message size	Time (ms)	Duty Cycle 1% (s)
	27	82,18	8,218
	158	348,42	34,842
	113	256,26	25,626
	222	481,54	48,154
	206	450,82	45,082
	13	51,46	5,146
	52	133,38	13,338
	126	286,98	28,698
	92	215,3	21,53
	13	51,46	5,146
TOTAL	1022	2357,8	235,78

Applying the aforementioned reductions, we would have the results in Table II.

In this section, we achieve a noticeable reduction in message size and the number of messages using LO-CoAP-EAP as opposed to CoAP-EAP. Using CBOR, we reduce the EAP-NOOB size, without losing information. Through these changes, we achieve the required reduction, so the largest message of all fits into the largest LoRaWAN frame (i.e., 222 Bytes). Overall, we reduce the exchange to 10 messages and a total of 1022 bytes for the whole exchange.

C. Theoretical Transmission Time

In terms of the time required to complete the exchange, since the largest message surpasses the LoRaWAN limit, we have to account for the final version with the optimisations of LO-CoAP-EAP and CBOR. Be it for the 222 Bytes limit if the transmissions are done with proxy, or 242 Bytes if they are done without a proxy. The theoretical values of the Time-On-Air are based on the Semtech tool [15]. Table III lists the configuration parameters of the tool, and Table IV lists the Time-On-Air values for the EAP-NOOB exchange. Analysing the latter table, we can see the transmission time for each message in ms and the duty cycle of 1% expected to wait after sending the message. We can observe that the time of transmission is $\approx$2.35 seconds and the waiting time due to duty cycle $\approx$235 seconds. Thus, we could run the EAP-NOOB exchange with LO-CoAP-EAP in less that 5 minutes.

While this time may be considered high, we have to consider that the bootstrapping process will take place only once when the device is turned ON, or when we need to do a re-bootstrapping. In such technologies, these values are not considered excessive.

V. Conclusions and Future work

Due to the popularity of the constrained devices in IoT, it requires adaptation of the existing bootstrapping processes to allow a secure joining of a new device to an existing and running network. To this end, we investigate a lightweight standalone solution to be applied to any IoT technology. In this paper, we reviewed one of the EAP methods. We considered the EAP-NOOB method, since it does not require to pre-establish the credentials on each device. We provide a theoretical analysis of the size reductions we can achieve in order to adjust EAP-NOOB in LoRaWAN. Therefore, we applied two distinct techniques to reduce the number of bytes. Firstly, we propose the use of an EAP lower layer design for the Internet of Things, LO-CoAP-EAP as opposed to CoAP-EAP. Secondly, we use CBOR to reduce the size of the messages sent over the air without losing information. By employing these changes, we achieved a reduction in size for all the messages. Thus, it allows all of them to fit into the largest LoRaWAN frame.

As for the future work, we plan to explore new techniques towards further reducing the size of the messages. Moreover, we plan to perform real experiments to validate our proposal and to evaluate how the different timers affect the entire authentication process, i.e. EAP-NOOB timers and *Waiting Exchange*, EAP protocol timers, AAA timers, and CoAP retransmission timers.

Acknowledgements

This work has been partially funded by the EU H2020 projects: Plug-n-Harvest with GA 768735, Fed4IoT with GA 814918, CYSEMA with GA 777455, and IoTCrawler with GA 779852. Moreover the the paper is is partially funded by THD GUARDIAN with GA TSI-100110-2019-20 as well as Grant DI-16-08432 of Industrial Doctorate from MINECO and PEANA UNMU13-2E-2536. Additionally, this work was partially performed and supported under the TPI ANR-17-CE10-0007-01 project of the French National Research Agency.

References

[1] J. Vollbrecht, J. D. Carlson, L. Blunk, D. B. D. A. Ph.D., and H. Levkowetz, “Extensible Authentication Protocol (EAP),” RFC 3748, Jun. 2004. [Online]. Available: https://rfc-editor.org/rfc/rfc3748.txt

[2] G. Gross, C. de Laat, D. Spence, L. H. Gommans, and J. Vollbrecht, “Generic AAA Architecture,” RFC 2903, Aug. 2000. [Online]. Available: https://rfc-editor.org/rfc/rfc2903.txt

[3] D. Garcia-Carrillo and R. Marin-Lopez, “Lightweight CoAP-Based Bootstrapping Service for the Internet of Things,” *Sensors*, vol. 16, no. 3, 2016. [Online]. Available: https://www.mdpi.com/1424-8220/16/3/358

[4] D. Garcia-Carrillo, R. Marin-Lopez, A. Kandasamy, and A. Pelov, “A CoAP-Based Network Access Authentication Service for Low-Power Wide Area Networks: LO-CoAP-EAP,” *Sensors*, vol. 17, no. 11, p. 2646, nov 2017. [Online]. Available: http://www.mdpi.com/1424-8220/17/11/2646

[5] T. Aura and M. Sethi, “Nimble out-of-band authentication for EAP (EAP-NOOB),” Internet Engineering Task Force, Internet-Draft draft-aura-eap-noob-04, 2018, Work in Progress.

[6] C. Bormann and P. E. Hoffman, “Concise Binary Object Representation (CBOR),” RFC 7049, Oct. 2013. [Online]. Available: https://rfc-editor.org/rfc/rfc7049.txt

[7] S. Farrell, “Low-Power Wide Area Network (LPWAN) Overview,” RFC 8376, May 2018. [Online]. Available: https://rfc-editor.org/rfc/rfc8376.txt

[8] A. Marquet, N. Montavont, and G. Z. Papadopoulos, “Investigating theoretical performance and demodulation techniques for LoRa,” in *Proceedings of the 1st International Workshop on Data Distribution in Industrial and Pervasive Internet (DIPI)*, 2019.

[9] ——, “Towards an SDR implementation of LoRa: reverse-engineering, demodulation strategies and assessment over Rayleigh channel,” *Elsevier Computer Communications*, vol. 153, pp. 595–605, 2020.

[10] D. Zorbas, G. Z. Papadopoulos, P. Maille, N. Montavont, and C. Douligeris, “Improving LoRa Network Capacity Using Multiple Spreading Factor Configurations,” in *Proceedings of the 25th International Conference on Telecommunication (ICT)*, 2018, pp. 516–520.

[11] S. Aguilar, A. Marquet, L. Toutain, C. Gomez, R. Vidal, N. Montavont, and G. Z. Papadopoulos, “LoRaWAN SCHC Fragmentation Demystified,” in *Proceedings of the 18th International Conference on Ad Hoc Networks and Wireless (AdHoc-Now)*, 2019.

[12] LoRa Alliance, Inc., “LoRaWAN 1.1 Regional Parameters.” 2017.

[13] A. Minaburo, L. Toutain, and R. Andreasen, “LPWAN Static Context Header Compression (SCHC) for CoAP,” Working Draft, IETF Secretariat, Internet-Draft draft-ietf-lpwan-coap-static-context-hc-13, March 2020, (Work in Progress).

[14] D. Garcia, R. Lopez, A. Kandasamy, and A. Pelov, “Lorawan authentication in radius,” Working Draft, IETF Secretariat, Internet-Draft draft-garcia-radext-radius-lorawan-03, May 2017, (Work in Progress).

[15] Semtech, “Semtech - SX1272 LoRa Calculator Tool,” Web, Semtech, Software, 2016, available Online (last access on March 2016) http://www.semtech.com.